

This "Listing of Claims" shall replace all prior versions, and listings, of claims in the application.

LISTING OF CLAIMS:

1. (Currently Amended) A software tool containing machine readable instructions stored on a physical medium for monitoring behavior of a running computer program for code patterns that violate a given set of coding rules, the software tool comprising:

a pattern detector manager including machine readable instructions for inserting into a running computer program a main entry breakpoint at one or more defined points in the computer program; and

a plurality of pattern detectors, each of the pattern detectors being associated with one of a group of defined coding patterns, each of the coding patterns including a plurality of steps, and each of said steps being associated with a defined location in the computer programs, and each of the pattern detectors further including machine readable instructions;

to insert additional breakpoints into the computer program to identify the steps in the group of defined coding patterns;

wherein, upon hitting one of the main entry breakpoints in the computer program, the pattern detector manager invokes each of the pattern detectors to insert the plurality of additional breakpoints into the computer program to identify said steps in the group of ~~defined coding patterns~~ at the plurality of defined locations in the computer program associated with the steps of the coding pattern associated with said each pattern detector, and the pattern detectors track the inserted, additional breakpoints to detect violations of said group of defined coding patterns.

Claims 2 and 3 (Cancelled).

4. (Previously Presented) A software tool according to Claim 1, for use with a debugger for debugging the computer program, and further including a launcher to invoke the pattern detector manager when the debugger is used to debug the program.

5. (Previously Presented) A software tool according to Claim 1, wherein the pattern detector manager removes the entry breakpoints at specified times.

6. (Previously Presented) A software tool according to Claim 1, wherein the pattern detector manager removes the entry breakpoints and the further breakpoints at specified times.

7. (Previously Presented) A software tool according to Claim 1, wherein:

the pattern detector manager includes means for monitoring for the occurrences of the entry breakpoints; and

the pattern detector manager inserts said at least one further breakpoint into the computer program in response to the monitoring means detecting the occurrence of said one of the entry breakpoints.

Claims 8, 9 and 10 (Cancelled).

11. (Previously Presented) A method according to Claim 21, for use with a debugger for debugging the computer program, and further including the step of invoking the pattern detector manager when the debugger is used to debug the program.

12. (Previously Presented) A method according to Claim 21, wherein further including the step of removing the entry breakpoints at specified times.

Claims 13 and 14 (Cancelled).

15. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for monitoring behavior of a running computer program, said method steps comprising:

using a pattern detector manager to insert into a running computer program a main entry breakpoint at one or more defined points in the computer program; and

using a plurality of pattern detectors for monitoring the computer program, wherein each of the pattern detectors is associated with one of a group of defined coding patterns, each of the coding patterns including a plurality of steps, and each of said steps being associated with a defined location in the computer program, including the step of inserting a plurality of additional breakpoints into the computer program to identify the steps in the group of defined coding patterns;

wherein, upon hitting one of the main entry breakpoints in the computer program, the pattern detector manager invokes each of the pattern detectors to insert the plurality of additional

breakpoints into the computer program ~~to identify said steps in the group of defined coding patterns~~ at the plurality of defined locations in the computer program associated with the steps of the coding pattern associated with said each pattern detector, and the pattern detectors track the inserted, additional breakpoints to detect violations of said group of defined coding patterns.

Claim 16 (Cancelled).

17. (Original) A program storage device according to Claim 15, for use with a debugger for debugging the computer program, and wherein said method steps include the further step of invoking the pattern detector manager when the debugger is used to debug the program.

18. (Original) A program storage device according to Claim 15, wherein said method steps include the further step of removing the entry breakpoints at specified times.

19. (Previously Presented) A program storage device according to Claim 15, wherein said method steps include the further step of removing the entry breakpoints and the further breakpoints at specified times.

Claim 20 (Cancelled).

21. (Currently Amended) A method of detecting code patterns in a computer program that violate a given set of coding rules, the method comprising the steps of:

defining a set of coding rules, each coding rule of the set of coding rules being associated with a respective one pattern detector of a set of pattern detectors, each coding rule of the set of coding rules including a plurality of steps, and each of said steps being associated with a defined location in the computer program;

providing a pattern detector manager for managing said pattern detectors;

providing a computer program, and running the computer program in a debug mode;

the pattern detector manager identifying, during the running of the computer program in the debug mode, points in the computer program that relate to said coding rules, and

said pattern detector manager inserting into the computer program an entry breakpoint at each of said identified points;

said pattern detector manager invoking each of the pattern detectors to monitor the computer program for a violation of the coding rule associated with said each of the pattern detectors, including the step of:

each of the pattern detectors inserting ~~one or more~~ a plurality of further breakpoints into the computer program ~~to identify further points~~ at the plurality of defined locations in the computer program ~~that relate to~~ associated with the steps of the coding rule associated with said each of the pattern detectors, and

tracking said additional breakpoints to determine whether the computer program violates the coding rule associated with said each of the pattern detectors, wherein each of said additional breakpoints identifies a respective step in the computer program that is part of the coding pattern associated with said one of the entry breakpoints, and wherein each of the pattern detectors monitors the computer program for the occurrence of any one of the first set of defined

conditions, the occurrence of which violates the coding rule associated with said each of the pattern detectors and monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors.

Claim 22 (Cancelled).

23. (New) The software tool according to Claim 1, wherein the plurality of additional breakpoints inserted into the computer program by each of the pattern detectors are inserted into the computer program sequentially in time.

24. (New) The software tool according to Claim 23, wherein:

for each of the pattern detectors, a first of the additional breakpoints is inserted into the computer program at the first time, and a second of the additional breakpoints is inserted into the computer program only when said first of the additional breakpoints is reached in the computer program; and

the pattern detectors detect said violations during execution of the computer programs and without stopping said execution.